Amendments to the Claims:

This listing of claims will replace all prior versions, and listings, of claims in the application:

Listing of Claims:

Please cancel claim 63 without prejudice.

Please amend claims 25, 26, 60, 66, and 73 as follows:

Claims 1-24 (canceled).

25. (currently amended): A method of condition generation comprising the steps of:

~~defining a set of arithmetic condition flags (ACFs);~~

~~specifying a condition code within~~receiving a first instruction having first and second operands containing packed data type elements, the packed data type elements in the first operand corresponding to the packed data type elements in the second operand and being associated with arithmetic condition flags (ACFs) from a set of ACFs;

executing ~~the~~ packed data operations specified in the first instruction on the corresponding packed data type elements;

~~updating~~setting the associated ACFs based upon the ~~specified condition code and a side effect resulting from the execution of the first instruction~~ results of the packed data operations;

determining whether to execute a further packed data operation on packed data type elements received in a second instruction ~~based on the state of the arithmetic condition flags~~, at least one packed data type element received in the second instruction being associated with an ACF from the set of ACFs; and

executing ~~the second instruction~~the further packed data operation on the at least one packed data type element if ~~it is determined to execute the second instruction~~the ACF associated with the at least one packed data type element is set.

26. (currently amended): The method of claim 25 wherein the first instruction includes a condition code to combine with the results of the packed data operations in setting the associated ACFs, the method further comprising the step of:

combining a previous state of the set of ACFs with the result of a condition code test specified by a current compare instruction to create a complex condition.

27. (original): The method of claim 25 wherein the condition code specifies a condition such as greater than (GT), less than (LT), equal (EQ) or less than or equal (LEQ).

28. (previously presented): The method of claim 26 wherein the compare instruction is further utilized to specify the desired conditions to be tested and two source registers to be compared.

29. (original): The method of claim 28 wherein the compare instruction is further utilized to specify a data type covering packed data forms.

30. (original): The method of claim 28 wherein the compare instruction is further utilized to specify a Boolean combination specification field.

31. (original): The method of claim 26 further comprising the step of controlling branching in a sequence processor (SP) based upon the created complex condition.

32. (previously amended): The method of claim 26 further comprising the step of conditionally executing in a sequence processor (SP) and at least one processing element (PE)

based on the created complex consisting of a Boolean combination of multiple conditions based upon the created complex condition.

33. (original): The method of claim 26 further comprising the step of conditionally executing on a combination of multiple conditions based upon the created complex condition.

Claims 34-57 (canceled).

58. (previously presented): The method of claim 26 wherein the condition code is chosen from the group consisting of a carry indication, an overflow indication, a sign indication, and a zero indication.

Claim 59 (canceled)

60. (currently amended): A method of conditional instruction execution comprising:

setting arithmetic scalar flags based on at least one side effect of the execution of a first instruction wherein the first instruction performs an operation on packed data comprising a plurality of data elements;

setting one arithmetic condition flags ~~based on the arithmetic scalar flags as specified by the first instruction~~ for each data element of the packed data;

determining whether to execute a second instruction based on the state of the arithmetic condition flags set by the first instruction; and

executing the second instruction if it is determined to execute the second instruction wherein the second instruction comprises at least one selectable conditional execution instruction opcode bit specifying the conditional execution of the second instruction.

61. (previously presented): The method of claim 60 wherein the arithmetic condition flags are set based on a Boolean combination of the arithmetic scalar flags.

62. (previously presented): The method of claim 60 wherein the execution of the second instruction affects the arithmetic scalar flags.

63. (canceled)

64. (previously presented): The method of claim 60 wherein the execution of the second instruction affects the state of the arithmetic condition flags.

65. (previously presented): The method of claim 60 wherein the first instruction is executed by a first processing element and the second instruction is conditionally executed by a second processing element.

66. (currently amended): An apparatus for conditional instruction execution comprising:

means for executing packed data operations specified in a first instruction, the first instruction having ~~one or more bits to indicate how to set~~ first and second operands containing packed data type elements, the packed data type elements in the first operand corresponding to the packed data type elements in the second operand and being associated with arithmetic condition flags (ACFs) of a set of ACFs;

means for setting associated arithmetic condition flags based on ~~said one or more bits and a side effect~~ results of the packed data operations ~~from the execution of the first instruction~~;

means for determining whether to execute a packed data operation on one or more packed data type elements specified in a second instruction ~~based on the state of the~~, at least one packed

data type element received in the second instruction being associated with arithmetic condition flags of a set of ACFs ~~set by the execution of the first instruction and said one or more bits~~; and

means for executing a packed data operation on at least one packed data type element~~the second instruction~~ if ~~it is determined to execute the second instruction~~the ACF associated with the at least one packed data type element is set.

67. (original): The apparatus of claim 66 wherein the first instruction is a compare instruction.

68. (previously presented): The apparatus of claim 67 wherein each of said one or more bits is chosen to indicate a condition from the group consisting of a carry indication, an overflow indication, a sign indication, and a zero indication.

69. (canceled)

70. (original): The apparatus of claim 66 wherein the second instruction comprises at least one selectable conditional execution instruction opcode bit specifying the conditional execution of the second instruction.

Claims 71-72 (canceled)

73. (currently amended): A processing apparatus for conditional instruction execution comprising:

a storage device for storing ~~an~~ set of arithmetic condition flags (ACFs);

an execution unit for ~~executing~~ receiving a first instruction, the first instruction having first and second operands containing packed data type elements, packed data type elements in the first operand corresponding to the packed data type elements in the second operand and being

associated with ACFs from the set of ACFs, ~~and~~the execution unit executing packed data operations specified in the first instruction, generating at least two condition states ~~as at least one side effect of the execution,~~ from results of the packed data operations; and

a generation unit for receiving the at least two condition states, ~~generating~~setting the associated ~~an~~ at least two arithmetic condition flags utilizing both ~~the~~a condition state ~~and an opcode bit from the first instruction~~, and storing the at least two arithmetic condition flags in the storage device, said execution unit ~~for~~ conditionally executing a second instruction based on the state of the at least two arithmetic condition flags.

74. (original): The processing apparatus of claim 73 wherein the first instruction is a compare instruction.

75. (original): The processing apparatus of claim 74 wherein the at least one side effect is chosen from the group consisting of a carry indication, an overflow indication, a sign indication, and a zero indication.

76. (canceled)

77. (original): The processing apparatus of claim 73 wherein the second instruction comprises at least one selectable conditional execution instruction opcode bit specifying the conditional execution of the second instruction.

78. (previously presented): The processing apparatus of claim 73 wherein the arithmetic condition flags are set based on a Boolean combination of the generated condition state.

79. (previously presented): The processing apparatus of claim 73 wherein the execution of the second instruction affects the generated condition state.